

VŠB - Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra informatiky

Absolvování individuální odborné praxe
Individual Professional Practice in the
Company

2011

Jan Salich

Zadání bakalářské práce

Student:

Jan Salich

Studijní program:

B2647 Informační a komunikační technologie

Studijní obor:

2612R025 Informatika a výpočetní technika

Téma:

Absolvování individuální odborné praxe
Individual Professional Practice in the Company

Zásady pro vypracování:

1. Student vykoná individuální praxi ve firmě: AutoCont CZ a.s.
2. Struktura závěrečné zprávy:
 - a) Popis odborného zaměření firmy, u které student vykonal odbornou praxi a popis pracovního zařazení studenta.
 - b) Seznam úkolů zadaných studentovi v průběhu odborné praxe s vyjádřením jejich časové náročnosti.
 - c) Zvolený postup řešení zadaných úkolů.
 - d) Teoretické a praktické znalosti a dovednosti získané v průběhu studia uplatněné studentem v průběhu odborné praxe.
 - e) Znalosti či dovednosti scházející studentovi v průběhu odborné praxe.
 - f) Dosažené výsledky v průběhu odborné praxe a její celkové zhodnocení.

Seznam doporučené odborné literatury:

Podle pokynů konzultanta, který vede odbornou praxi studenta.

Formální náležitosti a rozsah bakalářské práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí bakalářské práce: **Ing. Jan Gaura**

Konzultant bakalářské práce: Ing. Pavel Odstrčil

Datum zadání: 19.11.2010

Datum odevzdání: 06.05.2011



doc. Dr. Ing. Eduard Sojka
vedoucí katedry



prof. RNDr. Václav Snášel, CSc.
děkan fakulty

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

V Ostravě 4. května 2011

.....

Tímto bych chtěl poděkovat vedení společnosti AutoCont CZ a.s. za poskytnutí možnosti absolvování odborné praxe a celému kolektivu divize Vývoj softwaru za příjemnou spolupráci. Jmenovitě děkuji Lukášovi Kubisovi za mnoho cenných rad a pomoc při řešení zadaných úkolů.

Abstrakt

Tato bakalářská práce se zabývá popisem vykonávané pracovní činnosti v průběhu absolvování individuální odborné praxe ve firmě AutoCont CZ a.s. Je strukturována do tří hlavních částí, ve kterých se zabývám popisem společnosti, řešenými úkoly a hodnocením této studentské stáže. Jejím cílem je tak zrekapitulovat vykonávanou pracovní činnost pro tuto firmu v průběhu dvou semestrů třetího ročníku bakalářského studia.

Klíčová slova

Odborná praxe, AutoCont CZ a.s., vývoj softwaru, testování, UML, HTML, CSS, SQL, C#, .NET Framework, Unit Testing Framework, WPF.

Abstract

This bachelor thesis describes occupation performed during professional individual practice in the AutoCont CZ a.s. It is structured to three main parts, where I describe the company, tasks, and evaluation of the internship. The aim of this thesis is to summarize workload for the company during two semesters of the third year of undergraduate study.

Keywords

Professional practice, AutoCont CZ a.s., software development, testing, UML, HTML, CSS, SQL, C#, .NET Framework, Unit Testing Framework, WPF.

Seznam použitých symbolů a zkratek

BP	Business Process
CP	Centrální prodej
CPU	Central Processing Unit
CSS	Cascading Style Sheets
EA	Enterprise Architect
EBA	Enterprise Business Applications
ECM	Enterprise Content Management
ERP	Enterprise Resource Planning
GPU	Graphic Processing Unit
HTML	HyperText Markup Language
HTTP	Hypertext Transfer Protocol
ICT	Information and Communication Technologies
ISO	International Organization for Standardization
IT	Information technology
ITI	IT infrastruktura
ITSM	IT Service Management
MDČR	Ministerstvo dopravy České republiky
MS	Microsoft
PAS	Podnikové aplikace a služby
SQL	Structured Query Language
SW	Software
VSU	Vývoj softwaru
WPF	Windows Presentation Foundation

Obsah

1.	Úvod.....	1
2.	Odborné zaměření firmy a pracovní zařazení studenta	2
2.1	Popis odborného zaměření firmy	2
2.1.1	Divizní struktura společnosti.....	2
2.2	Popis pracovního zařazení studenta	3
3.	Zadané úkoly.....	4
3.1	Studium metodik analýzy a testování	4
3.2	Úprava webových stránek a práce s databází MS SQL Server	4
3.3	Testování aplikace výrobní linky KC59.....	5
4.	Řešení zadaných úkolů.....	6
4.1	Studium metodik analýzy a testování	6
4.1.1	Metodika analýzy	6
4.1.2	Metodika testování	6
4.1.3	Praktické ověření teorie.....	8
4.2	Úprava webových stránek a práce s databází MS SQL Server	9
4.2.1	Úprava webových stránek	9
4.2.2	Práce s databází MS SQL Server	10
4.3	Testování aplikace výrobní linky KC59.....	10
4.3.1	Testování jednotek	10
4.3.2	Analýza chování aplikace	12
5.	Uplatněné znalosti a dovednosti získané v průběhu studia	14
6.	Scházející znalosti a dovednosti v průběhu praxe.....	15
7.	Závěr a celkové hodnocení odborné praxe.....	16
8.	Literatura.....	17

1. Úvod

Cílem této práce je sestavit závěrečnou zprávu o mém působení na odborné praxi ve firmě AutoCont CZ a.s., kde jsem pracoval jako junior programátor a tester. Účastnil jsem se vybraných firemních projektů, ve kterých jsem uplatňoval znalosti nabyté v průběhu studia, rozšiřoval dovednosti v programování a učil se novým postupům a řešením v oblasti vývoje softwaru. Plněné úkoly se zabývaly databázemi, úpravami webových stránek a testováním softwaru. Nejdéle jsem byl zaměstnán psaním jednotkových testů v rámci projektu týkajícího se vytvoření nové výrobní linky na kompresory KC59 pro společnost Valeo Compressor Europe, s.r.o.

První kapitola je zaměřena na seznámení se s historickým vývojem, strukturou a odborným zaměřením této společnosti. Popisují zde své pracovní zařazení a jeho náplň.

V další kapitole se věnuji zadání úkolů, které jsem v průběhu praxe dostal. Abych se mohl pustit do jejich plnění, musel jsem nejdříve nastudovat vybrané metodiky týkající se vývoje softwaru. Díky nim a nabytým znalostem ze studia jsem mohl započít své působení plněním různých úkolů. Postupem času rostla mírně i jejich obtížnost, dle které jsou v tomto dokumentu úkoly vzestupně seřazeny. Zadání jsou rozdělena do tří částí. V první z nich popisují studované metodiky, další obsahuje menší všeobecné úkoly týkající se MS SQL databáze a webových stránek. Třetí a největší část je věnována testování softwaru.

Ve třetí kapitole se podrobněji zabývám postupem řešení jednotlivých úkolů. Každé řešení obsahuje větší či menší teoretický základ, díky kterému bude snazší proniknout do problematiky daného úkolu.

Bez znalostí a dovedností získaných v průběhu studia na vysoké škole, bych se na odborné praxi neobešel. V této kapitole jsou vybrány předměty a učivo ze všech tří semestrů studia, které mně bylo nápomocné při plnění úkolů.

V předposlední kapitole jsou uvedeny znalosti a dovednosti, které mi na praxi scházely, a musel jsem se je samostatně doučit. Dle přiměřeně zvolené obtížnosti řešených úkolů jich není mnoho.

Závěrem uvádím celkové zhodnocení odborné praxe a výsledky, kterých jsem dosáhl.

2. Odborné zaměření firmy a pracovní zařazení studenta

2.1 Popis odborného zaměření firmy

AutoCont CZ a.s. [1] je českou soukromou společností, která v oblasti dodávek zboží, služeb a řešení na poli informačních a komunikačních technologií úspěšně působí již 20 let. K rychlému rozvoji v prvních 10 letech přispěla výroba vlastní značky osobních počítačů a systematické budování rozsáhlé obchodní a servisní sítě v České republice a na Slovensku. V druhém desetiletí své existence se AutoCont zaměřil na vybudování specializovaných divizí a postupně se pracoval mezi nejvýznamnější firmy i v oblasti rozsáhlých a složitých ICT projektů.

Nabídka společnosti AutoCont je v oblasti informačních a komunikačních technologií (ICT) zaměřena do těchto hlavních směrů: IT infrastruktura, outsourcing, podnikové aplikace a služby, vývoj softwaru, systémová integrace a centrální prodej. Dle těchto směrů je rozdělena i struktura společnosti do stejnojmenných divizí.

2.1.1 Divizní struktura společnosti

➤ IT infrastruktura

Předmětem činnosti divize IT infrastruktura (ITI) je poskytování specializovaných a profesionálních služeb spojených s celým životním cyklem IT infrastruktury. Od prosté dodávky IT zařízení a technologií, přes návrh a realizaci infrastrukturních řešení, až po správu a podporu následného provozu.

➤ Outsourcing

Divize IT service management (ITSM) se zaměřuje na dodávky služeb v oblasti provozu informačních systémů u velkých zákazníků. Těžiště nabízených služeb spočívá v provádění správy a údržby informačního systému zákazníka nebo jeho části. Nejvyšší úroveň nabízených služeb je forma kompletního outsourcingu provozu informačního systému včetně převzetí majetku a pracovníků.

➤ Podnikové aplikace a služby

Hlavní činností divize Podnikové aplikace a služby (PAS) je implementace, podpora a rozvoj podnikových informačních systémů. Jedná se o vzájemně propojené informační systémy, složené z aplikací pro ekonomiku, plánování a řízení zdrojů, tzv. Enterprise Resource Planning (ERP) systémy, dále informační systémy pro kolektivní spolupráci, organizaci práce a komunikaci a manažerské informační systémy.

➤ **Enterprise Business Applications**

Divize Enterprise Business Applications (EBA) se zabývá správou podnikového obsahu, tzv. Enterprise Content Managementem (ECM), a řešením na platformě SAP. K jejím zákazníkům patří zejména korporace a velké organizace státní nebo veřejné správy. Vedle nich ale jejich služeb a řešení využívá také řada dalších společností se specifickými požadavky na řešení v oblasti ERP, správy a oběhu dokumentů včetně jejich skenování, podpory práce s formuláři případně podpory interních nebo zákaznických procesů.

➤ **Vývoj softwaru**

Divize Vývoj softwaru (VSW) se zabývá poskytováním komplexních služeb v oblasti vývoje softwaru na zakázku. Přináší mnoha firmám a organizacím konkurenční výhodu prostřednictvím řešení stavěných na míru pro konkrétní klienty, řešení s vysokou přidanou hodnotou, kterou nemohou získat nákupem standardních produktů.

➤ **Centrální prodej**

Divize Centrální prodej (CP) se primárně zaměřuje na malé organizace, které často nemají své vlastní IT specialisty. Pro podnikatele, malé firmy, města, obce, školy a další vzdělávací zařízení nabízí rychlé dodávky software, hardware a moderní IT služby.

2.2 Popis pracovního zařazení studenta

Po nástupu na odbornou praxi jsem se stal součástí týmu divize VSW a byla mně přidělena pozice junior programátora a testera. V pracovním týmu divize se aktuálně pracovalo na několika interních projektech, byl jsem zařazen do následujících z nich.

➤ **IT Dnes**

Projekt IT Dnes se zabýval inovací portálu IT Dnes zaměřeného na prodej software, hardware a IT služeb. Tento e-shop provozuje společnost AutoCont CZ a.s. Při tomto projektu jsem se věnoval úpravě webových stránek portálu společnosti.

➤ **eTesty MDČR**

Tento projekt se zabýval vývojem aplikace eTesty Ministerstva dopravy České republiky (MDČR) používající se k testování znalostí uchazečů o řidičské oprávnění na Obecních úřadech a v autoškolách. Zde jsem se prováděl dotazování na MS SQL databázi eTestů MDČR.

➤ **Valeo**

Projekt Valeo se zabývá implementací softwaru pro výrobní linku KC59 společnosti Valeo Compressor Europe, s.r.o. Působil jsem při něm jako tester softwaru. Mým úkolem bylo pomocí unit testů, či jiných dostupných prostředků otestovat funkčnost aplikace.

3. Zadané úkoly

V průběhu odborné praxe jsem dostával zadání různých úkolů a jejich řešením přispíval k efektivnější týmové spolupráci. Abych mohl svým kolegům pomoci, musel jsem nastudovat zadání a částečně porozumět řešení projektů, na kterých v týmu pracovali.

První dny na praxi patřily seznamování se s kolektivem a prostředím firmy. Softwarové vybavení mně bylo až na výjimky známé ze studia a z osobního života. Po tomto sblížení jsem průběžně dostával zadání následujících úkolů.

3.1 Studium metodik analýzy a testování

Úkolem bylo prostudovat metodiky analýzy a testování, které jsou procesním a odborným podkladem pro oddělení divize Vývoj softwaru (VSW). Pomocí nástroje Enterprise Architect (EA) jsem si měl nabytou teorii ověřit prakticky. Konkrétně se jednalo o pochopení účelu namodelovaného systému automatizace výrobní linky pomocí diagramů třídních, aktivitních a diagramů užití. Cílem studia metodik nebylo se je učit zpaměti, ale seznámit se s vývojem software a jeho řízeným rozvojem.

3.2 Úprava webových stránek a práce s databází MS SQL Server

První úkol se týkal zpřehlednění kódu v hypertextovém značkovacím jazyce (HTML) a úpravy vzhledu webových stránek e-shopu společnosti AutoCont CZ a.s. Pro grafickou úpravu webových stránek se používají kaskádové styly (CSS) a využívají se hlavně při správě většího webu s mnoha stránkami, které by měly vypadat podobně [2]. Jsou známy tři metody zápisu CSS: přímo k elementům HTML, do hlavičky HTML dokumentu a do externího CSS souboru [3]. Při formátování webových stránek e-shopu společnosti bylo využíváno všech tří způsobů dohromady. Úkolem bylo sjednotit způsob zápisu kaskádových stylů s využitím pouze externího CSS souboru. Po kompletním oddělení informačního obsahu webových stránek od jejich grafického designu následovala změna spektra používaných barev při jejich formátování. Cílem této části úkolu bylo připravit webové stránky na jejich kompletní změnu vzhledu (redesign).

Ve druhém úkolu obsahující práci s databází MS SQL Server bylo úkolem vyhledat požadované informace v obsáhlé databázi systému eTesty Ministerstva dopravy České republiky. K hledání měly být použity složené SQL dotazy používané pro výběr zobrazovaných dat. Cílem bylo zjistit časy vložení či smazání záznamu techniky a jízdy daným uživatelem a kolegům tak pomoci s ověřením pravosti údajů.

3.3 Testování aplikace výrobní linky KC59

Hlavní náplní tohoto úkolu bylo pomocí automatizovaných testů testovat jednotlivé funkce aplikace zajišťující evidenci a průchod komponent linky KC59 pro společnost Valeo Compressor Europe, s.r.o. Aplikace je založena na platformě Microsoft (MS) .NET Framework 3.5, která poskytuje aplikační prostředí, standardní funkcionalitu a programovou základnu. Vyvíjena byla v integrovaném prostředí MS Visual Studio 2010. Součástí zadání byl stále se rozšiřující testovací scénář obsahující několik za sebou jdoucích testovacích případů. Testovací případ je jeden nebo seznam kroků sloužící k testování chování a funkce systému při konkrétní situaci, která může při používání testované položky nastat. Přepsáním testovacího scénáře do programovacího jazyka vzniká testovací skript, který představuje automatické testy vyvíjené aplikace. Automatický test je prováděn softwarem a používá se při opakovaném spouštění velkého množství testů nebo testu s velkým množstvím generovaných dat [4]. Konkrétním příkladem automatického testování je testování jednotek (unit testing). Dobrý jednotkový test (unit test) je Royem Osherovem [5] definován slovy: „*Unit test je automatizovaný kus kódu, který vyvolá testovanou metodu nebo třídu a pak zkontroluje některé předpoklady logického chování této metody nebo třídy. Unit test je téměř vždy napsán pomocí unit-testing frameworku, díky kterému je napsán snadno a rychle běží. Unit test je plně automatizovaný, důvěryhodný, čitelný a udržitelný.*“

Cílem první části tohoto úkolu bylo na základě zadávaných testovacích případů vytvářet jednotkové testy a ověřovat jimi funkčnost nejmenších testovatelných jednotek napsaného softwaru (např. modulů, programů, objektů, tříd). Případné nalezení chyby bylo hlášeno architektovi tohoto projektu.

Další částí úkolu bylo otestování této aplikace za účelem analýzy jejího chování při běhu pomocí sady nástrojů MS WPF Performance Suite. Cílem bylo zjistit, které grafické prvky zpomalují běh aplikace, následně se pokusit dané vytížení redukovat a optimalizovat tak výkon aplikace.

4. Řešení zadaných úkolů

4.1 Studium metodik analýzy a testování

4.1.1 Metodika analýzy

Metodika analýzy [6] popisuje aktivity a předměty, které souvisí s analýzou vývojových projektů, s vnitřní organizací analytického oddělení v divizi Vývoj softwaru (VSW), organizací součinnosti analytického oddělení s vnitrofiremním okolím a s komunikací analytika se zákazníkem.

Účelem této metodiky je:

- **Zavést UML do oddělení vývoje**

Zavést standard Unified Modeling Language (UML) [7] znamená zavést přesnější normy týkající se projektů v analytické fázi, aby jejich výstupem mohl být popis „rozhraní“, které analýza poskytuje svým uživatelům (zákazníci, návrháři, programátoři, grafici, testéři, obchodníci či manažeři).

- **Sjednotit rozsah a detail analytických modelů**

Standard UML je velmi dobře srozumitelný a relativně jednoduchý, nedává však žádné pokyny ohledně použití množství různých diagramů a analytických modelů. Stejně tak UML nestanovuje, na jaké úrovni podrobnosti se má model zastavit. Sjednocení užití UML má ve firmě velmi příznivé účinky. Každý čtenář pak ví, kde a co může v modelu najít. Stejně typy elementů jsou v různých modelech přibližně stejně velké, proto lze počet analytických tříd a typových úloh používat jako projektové metriky a odhadovat tak například pracnost vývoje.

Uvedené cíle jsou prostředkem pro zefektivnění, zkvalitnění a změřitelnost procesů vývoje software a jeho řízeného rozvoje. Při studiu této metodiky jsem se s většinou informací setkal poprvé. Výjimku tvořila část věnující se UML diagramům.

4.1.2 Metodika testování

Metodika testování [8] popisuje procesy, které souvisí s testováním vývojových projektů, s vnitřní organizací testovacího oddělení v divizi Vývoj softwaru (VSW) a s organizací součinnosti testovacího oddělení s vnitrofiremním okolím.

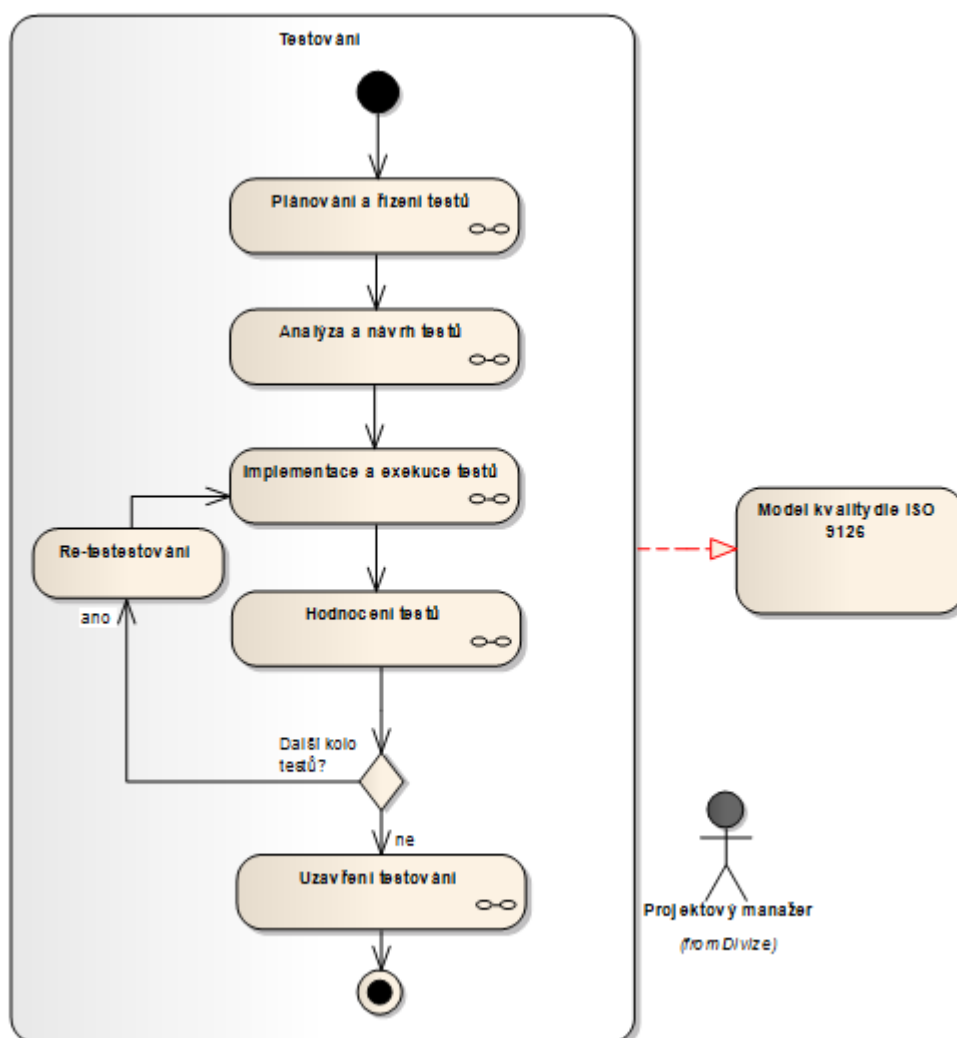
Účelem této metodiky je:

➤ **Zavést zjišťování kvality testování software dle ISO 9126 v divizi VSW**

Mezinárodní standard pro vyhodnocování kvality SW ISO 9126-1 popisuje tyto aspekty vyvíjeného produktu: funkčnost, použitelnost, výkonnost, spolehlivost, přenositelnost a udržitelnost. Jejich seznam je důležitý hlavně proto, aby nebylo opomenuto sesbírat požadavky a připravit testy pro žádné z těchto kritérií.

➤ **Podporovat zajišťování kvality v oddělení vývoje**

Zajišťování kvality SW je metodice testování popisováno v několika kapitolách, které odpovídají jednotlivým stavům UML diagramu aktivit Jany Hrotekové [8], autorky dokumentu.

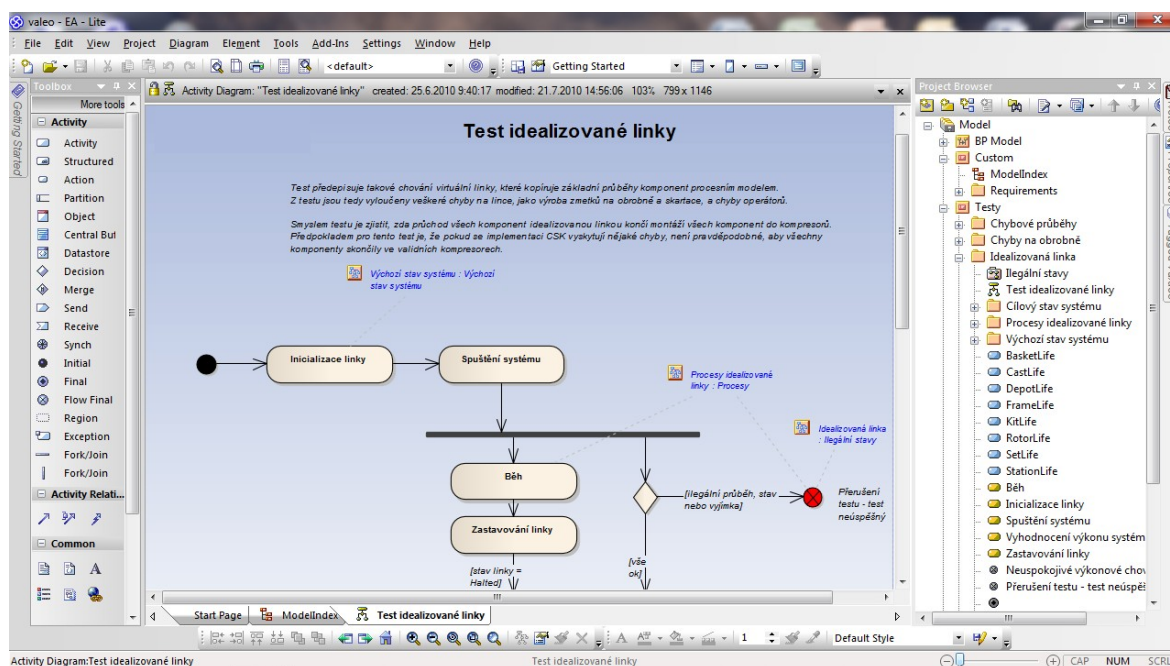


Obr. 1 – Testování v kontextu života projektu

Tato metodika stanovuje cestu pro sjednocení chápání účelů testování i v širším prostředí divize - od oblasti obchodních aktivit až do procesů podpory a rozvoje. Jejím studiem jsem se seznámil s velkým množstvím informací, které mně byly nápomocné při řešení úkolů týkajících se testování softwaru.

4.1.3 Praktické ověření teorie

K praktickému ověření získaných znalostí z obou metodik jsem využíval softwarový nástroj Enterprise Architect (EA) (Obr. 2 - Ukázka nástroje EA) australské společnosti Sparx Systems [9]. Tento software slouží k vytváření UML diagramů a podporuje celý životní cyklus vývoje softwaru. Využívá se pro podporu týmových softwarových projektů, pro analýzu, vytváření návrhů, dokumentaci, podporu managementu, podporu transformací modelů a podporu vizuální komunikace v týmu. Software Enterprise Architect je komplexním nástrojem a pokrývá všechny fáze vývoje software - od sběru požadavků přes návrh a generování kódu až k nasazení hotového díla na produkční prostředí včetně projektového řízení. Jako praktickou ukázkou jsem dostal vytvořený projekt obsahující analytický model k projektu Valeo - automatizace výrobní linky KC59, na kterém se v týmu aktuálně pracovalo. Pomocí této ukázky zahrnující business process (BP) model, doménové modely, modely případů užití a aktivní diagramy automatizovaných testů bylo snazší se zorientovat ve všech procesech a funkcích tohoto projektu, jehož aplikaci jsem měl v dalším průběhu testovat.



Obr. 2 - Ukázka nástroje EA

4.2 Úprava webových stránek a práce s databází MS SQL Server

4.2.1 Úprava webových stránek

Ze zdrojového kódu webových stránek v hypertextovém značkovacím jazyce (HTML) bylo patrné, že jejich grafická podoba od obsahu je již částečně oddělena s využitím externího Cascading Style Sheets (CSS) souboru. Na mně bylo najít v HTML dokumentu zbývající grafické úpravy elementů a rozumně je s využitím tříd (class) a identifikátorů (id) přenést do CSS souboru. Toto oddělení grafiky od obsahu mělo své důvody a výhody. Jelikož nově byl v jednom (HTML) dokumentu pouze obsah stránky a v druhém (CSS) dokumentu byly definice vzhledu stránek, vyznat se v kódu bylo mnohem přehlednější [3]. Webový prohlížeč si navíc může soubor se styly uložit do cache paměti a tím urychlit načítání obsahu webových stránek. Na druhou stranu dochází k dalšímu Hypertext Transfer Protocol (HTTP) požadavku navíc, než kdyby bylo použito psaní stylů přímo k elementům HTML nebo do hlavičky HTML dokumentu [10]. Případnou změnu vzhledu všech stránek webu je nyní možno jednoduše a rychle provést úpravou jednoho pravidla v externím CSS souboru. První část tohoto úkolu se mi podařilo zdárně dokončit.

Ve druhé části jsem se věnoval úpravě vzhledu webových stránek, konkrétně jsem všechny prvky zvýrazněné modrou barvou měnil na odpovídající barvu z červeného spektra. Podílel jsem se tak na kompletní změně jejich vzhledu (Obr. 3 - Redesign webových stránek). S využitím kaskádových stylů načítaných z externího souboru byl tento úkol za chvíli hotov.



Obr. 3 - Redesign webových stránek

4.2.2 Práce s databází MS SQL Server

Databáze k aplikaci eTesty Ministerstva dopravy České republiky (MDČR) byla vytvořena na databázové platformě MS SQL Server 2005. Aplikace slouží k ostrému i cvičnému testování znalostí uchazečů o řidičské oprávnění na Obecních úřadech a v autoškolách. Databáze byla proto velmi obsáhlá. Pro práci s ní jsem používal vývojové prostředí MS SQL Server Management Studio. Pomocí něj jsem si nejprve získal představu o struktuře databáze. Studoval jsem vazby mezi tabulkami definované pomocí primárních a cizích klíčů, atributy tabulek i jejich samotný obsah. Po této fázi jsem začal na databázi provádět SQL dotazy realizující operace projekce a selekce. Vybírány tedy byly pouze požadované sloupce a záznamy, které vyhovovaly zadané podmínce. U některých dotazů jsem využíval i klauzule JOIN, pomocí které jsem na základě odpovídajících si klíčů spojoval více tabulek. Pokud byl SQL dotaz zadán správně a v databázi se nacházely hledané informace, výsledek byl po spuštění dotazu vypsán na konzoli SQL Server Management Studia.

4.3 Testování aplikace výrobní linky KC59

4.3.1 Testování jednotek

Vyvinený systém pro automatizaci výrobní linky KC59 se používá v kritických místech, kde případné selhání aplikace může způsobit velké materiální škody. K zajištění spolehlivosti aplikace provádíme v procesu vývoje softwaru několik typů testů. Těmi nejzákladnějšími jsou testy jednotek (unit testy). Sada těchto automatizovaných testů zajišťuje otestování funkčnosti programu bez toho, abychom aplikaci sami spouštěli a zkoušeli v ní používat různé ovládací prvky. Jelikož aplikace byla vyvíjena v MS Visual Studio 2010, administrace testů byla velmi snadná. V této verzi Visual Studia je totiž obsažen Unit Testing Framework [11] podporující unit testy.

Obsahem mé práce v tomto úkolu bylo dle testovacích scénářů tvořených kombinací testovacích případů a UML vývojových diagramů vytvářet unit testy ověřující funkčnost a chování této aplikace. V již existujících testovacích třídách jsem vytvářel testovací metody, které používaly třídu nebo třídy z testované aplikace a prováděly otestování korektnosti jejich metod. Jednou z nich byla například následující testovací metoda `Get_FindByCast2D_ReturnCast` ověřující existenci odlitku v databázi podle evidenčního čísla.

```
[TestMethod]
[TestCategory("CastRepository")]
public void Get_FindByCast2D_ReturnCast()
{
    // arrange
    CastRepository target = new CastRepository();
    string cast2D = "F241 0000001";
```

```

// act
var actual = target.Find(cast2D);

// assert
Assert.IsNotNull(actual);
Assert.AreEqual(actual.Cast2D, cast2D);
}

```

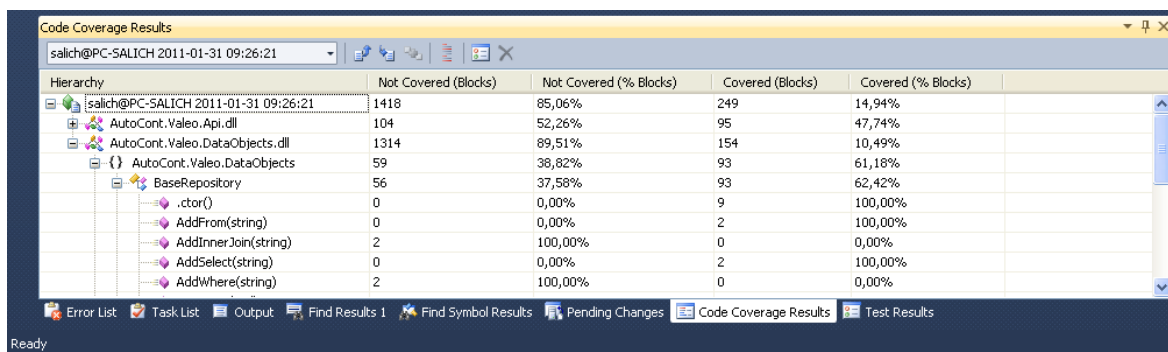
Tato metoda je umístěna v testovací třídě spolu s ostatními testovacími metodami. Každá takováto třída a metoda musí být označena atributem `TestClass`, resp. `TestMethod` [11]. Volitelně mohou být testovací metody umístěny do specifikované kategorie unit testů pomocí atributu `TestCategory`. Aby byla testovací metoda co nejpřehlednější, často se pomyslně rozděluje do tří částí. Stejně je tak tomu u této konkrétní metody. V její první části (arrange) se vytváří instance třídy `CastRepository` a definuje se proměnná `cast2D` včetně inicializace. V další části testu (act) se pomocí definované metody `Find` v databázi vyhledá odlitek podle evidenčního čísla. V třetí části (assert) se pomocí metod definovaných v třídě `Assert` ověří existence specifikovaného odlitku v databázi. Nejčastějším účelem vytvářených testovacích metod v tomto úkolu bylo porovnávat výsledky testovaného kódu s přepočítanými, ověřovat si zapsané hodnoty v databázi nebo očekávat zachycení cíleně vyvolané výjimky. K porovnávání výsledků bylo využíváno statické třídy `Assert` [11] obsahující metody ověřující, zda testovaná podmínka byla splněna nebo ne. Tato třída je definována ve jmenném prostoru `Microsoft.VisualStudio.TestTools.UnitTesting` umístěném v Unit Testing Frameworku [11]. Když jsem měl jeden nebo více testů hotových, spustil jsem je a čekal na jejich doběhnutí. Poté byla zobrazena tabulka s výsledky (Obr. 4 - Tabulka výsledků proběhlých testů). Každý test ideálně dával dvě hodnoty, buď byl úspěšný a prošel (passed) nebo úspěšný nebyl a selhal (failed). Pokud nějaký test neprošel, z detailů výsledku jsem zjistil, proč došlo k selhání, test opravil nebo nahlásil chybu architektovi projektu.

Result	Test Name	Project	Error Message
Passed	Test28	AutoCont.Valeo.SqlClr.Tests	
Passed	Test29	AutoCont.Valeo.SqlClr.Tests	
Passed	Test30	AutoCont.Valeo.SqlClr.Tests	
Passed	Test31	AutoCont.Valeo.SqlClr.Tests	
Passed	Test32	AutoCont.Valeo.SqlClr.Tests	
Passed	Test33	AutoCont.Valeo.SqlClr.Tests	
Passed	Test34	AutoCont.Valeo.SqlClr.Tests	
Passed	Test73	AutoCont.Valeo.SqlClr.Tests	
Passed	Test74	AutoCont.Valeo.SqlClr.Tests	

Obr. 4 - Tabulka výsledků proběhlých testů

K vykonaným testům je možné ve Visual Studiu si nechat zobrazit pokrytí kódu testových tříd (Obr. 5 - Tabulka pokrytí kódu testy). Ve zvláštním okně je pak ve stromové struktuře zobrazeno procentuální pokrytí pro jednotlivé assemblies, namespace, třídy a metody. Čím vyšší je hodnota

pokrytí kódu testy, tím vyšší je jejich kvalita. Se stále narůstajícím počtem testovacích scénářů jsem se testování věnoval největší část odborné praxe.



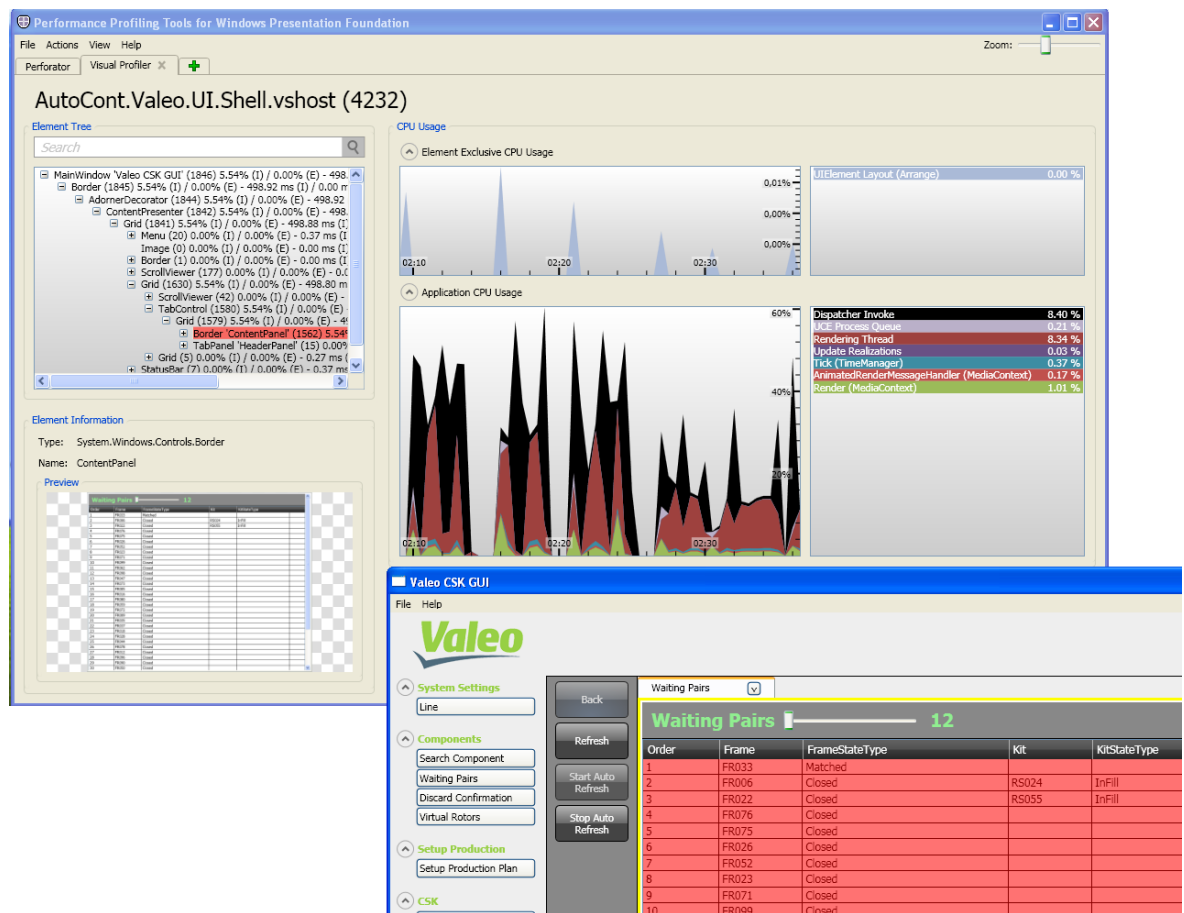
Hierarchy	Not Covered (Blocks)	Not Covered (% Blocks)	Covered (Blocks)	Covered (% Blocks)
salich@PC-SALICH 2011-01-31 09:26:21	1418	85,06%	249	14,94%
AutoCont.Valeo.Api.dll	104	52,26%	95	47,74%
AutoCont.Valeo.DataObjects.dll	1314	89,51%	154	10,49%
AutoCont.Valeo.DataObjects	59	38,82%	93	61,18%
BaseRepository	56	37,58%	93	62,42%
.ctor()	0	0,00%	9	100,00%
AddFrom(string)	0	0,00%	2	100,00%
AddInnerJoin(string)	2	100,00%	0	0,00%
AddSelect(string)	0	0,00%	2	100,00%
AddWhere(string)	2	100,00%	0	0,00%

Obr. 5 - Tabulka pokrytí kódu testy

4.3.2 Analýza chování aplikace

Testovací aplikace byla vytvořena pomocí technologie Windows Presentation Foundation (WPF). Taková to aplikace s uživatelsky bohatým rozhraním je pak renderována na grafické kartě a tím se ulehčuje práce procesoru počítače. V závislosti na grafickém hardwaru může nastat situace, kdy je aplikace částečně renderovaná na grafickém procesoru (GPU) a částečně na procesoru (CPU). V nejhorším případě bude aplikace renderována na CPU celá. Tehdy dojde k značnému zpomalení aplikace a počítače [12]. Aby testovaná aplikace byla co nejvíce nezávislá na použitém hardwaru, pomocí sady nástrojů WPF Performance Suite [13] jsem podrobně sledoval její náročnost na vykreslování a zjišťoval možné způsoby optimalizace jejího výkonu. WPF Performance Suite obsahuje dva nástroje analyzující výkon aplikace, Perforator a Visual Profiler. Perforator umožňuje podrobně analyzovat konkrétní chování vykreslovaných částí aplikace a Visual Profiler analyzuje výkon WPF služeb (např. layoutu, renderování a animace prvků). S pomocí druhého nástroje je pak snadné zjistit, který vizuální prvek může v testované aplikaci způsobovat potíže ve výkonu.

V rámci tohoto úkolu jsem nejčastěji využíval nástroj Visual Profiler, který je zobrazen spolu s oknem spuštěné testované aplikace na obrázku na další straně (Obr. 6 - Visual Profiler). Uživatelské rozhraní tohoto nástroje je rozděleno do několika částí. Vlevo je nahoře zobrazena stromová struktura elementů testované aplikace a dole je ukázán aktuálně označený prvek. V pravé části jsou rozloženy grafy znázorňující využití CPU celou aplikací a zvlášť vybraným elementem. Při spuštění analýzy je aktivní prvek zvýrazněn barvou z červeného spektra, která odpovídá velikosti vytížení CPU. Při používání aplikace jsem dle analýzy zjistil, že nejvíce výkonu spotřebuje výpis informací velkého množství informací do ovládacího prvku DataGridView, který je určen pro zobrazování výstupu z nějakého datového zdroje ve formátu tabulky. Řešením by mohla být virtualizace šablony DataGridViewu. Vytvářeny by tak byly pouze buňky, které jsou skutečně vidět na obrazovce a ostatní by byly vytvořeny na požádání například při posouvání okna [14]. V ostatních případech se aplikace jevila stabilní a nenáročná na výpočetní a grafický výkon.



Obr. 6 - Visual Profiler

5. Uplatněné znalosti a dovednosti získané v průběhu studia

Po celou dobu odborné praxe jsem využíval znalostí získaných ze všech tří ročníků studia na vysoké škole. Jedním z nejvíce nápomocných předmětů pro mě byl Úvod do informačních technologií, kde jsem získal základní znalosti týkající se programovacích jazyků, databází, vývoje softwaru i tvorby webových stránek. Tyto nabyté informace byly v průběhu dalšího studia rozšiřovány prostřednictvím specializovaných předmětů. Dovednosti programovat v jazyce C# vděčím předmětu Programovací jazyky 2. S databázemi umím pracovat hlavně prostřednictvím předmětů Teorie zpracování dat a Databázové informační systémy. Nebýt předmětu Softwarové inženýrství, neměl bych představu o vývoji rozsáhlých softwarových systémů a při něm používaném komunikačním prostředku, jazyku UML. Posledním nejvíce upotřebeným předmětem při praxi byl Vývoj internetových aplikací.

6. Scházející znalosti a dovednosti v průběhu praxe

Informační obsah a zaměření předmětů bylo na vysoké škole natolik rozsáhlé, že jsem se s velkými nedostatky svých znalostí na praxi nesetkal. Svůj vliv na to měla i obtížnost řešených úkolů, která byla řešena velice přiměřeně a měla vzestupný charakter. Několik scházejících znalostí a dovedností se v průběhu praxe přeci jen našlo. Na začátku to byla dovednost pracovat v rozsáhlém pracovním týmu a prosadit se v něm. Scházely mi také teoretické a praktické znalosti z oblasti testování softwaru.

7. Závěr a celkové hodnocení odborné praxe

Již před nástupem na odbornou praxi jsem věděl, že oproti klasické bakalářské práci má mnoho výhod. Jednou z nich byla možnost si v týmu zkušených pracovníků rozšířit a prakticky vyzkoušet studiem nabyté znalosti. Dále jsem chtěl touto cestou získat mnoho cenných rad a zkušeností, které mne budou nápomocné při studiu i v budoucím zaměstnání. Všechna má očekávání byla splněna. Se všemi náležitostmi jsem byl zařazen do týmu profesionálních vývojářů a s jejich pomocí se zdokonalil v programovacích jazycích, naučil se týmové spolupráce i pracovat s novými softwarovými nástroji. Díky jim jsem také poznal vnitřní fungování firmy zabývající se IT technologiemi. Jsem velmi rád, že jsem si vybral praxi právě ve společnosti AutoCont CZ a.s. a mohl být po celý školní rok její součástí.

8. Literatura

1. AutoCont CZ a.s. - Základní informace. [Online] [Citace: 1. květen 2011.] Dostupné z: <http://www.autocont.cz/profil-informace.cml>.
2. Jak psát web - Kaskádové styly. [Online] [Citace: 1. květen 2011.] Dostupné z: <http://www.jakpsatweb.cz/css/>.
3. Tvorba WWW stránek - Kam psát CSS. [Online] [Citace: 1. květen 2011.] Dostupné z: <http://www.stranektvorba.cz/kam-psat-css.html>.
4. **Borovcová, Anna.** *Testování webových aplikací, Část II: Základy testování.* [PDF] 2009.
5. **Osheroove, Roy.** *The Art of Unit Testing: with Examples in .NET.* Greenwich : Manning Publications Co., 2009. ISBN: 978-1-933988-27-6.
6. **Teocharisová, Vanda.** *Metodika analýzy.* [Dokument] Brno : AutoCont CZ a.s., 2010.
7. Unified Modeling Language. [Online] [Citace: 1. květen 2011.] Dostupné z: <http://www.uml.org/>.
8. **Hroteková, Jana.** *Metodika testování.* [Dokument] Brno : AutoCont CZ a.s., 2010.
9. Sparx Systems - Enterprise Architect. [Online] [Citace: 1. květen 2011.] Dostupné z: <http://www.sparxsystems.com.au/>.
10. About.com - External CSS Makes Faster Pages. [Online] [Citace: 1. květen 2011.] Dostupné z: <http://webdesign.about.com/od/speed/qt/use-one-external-css-file.htm>.
11. MSDN Library - Unit Testing Framework. [Online] [Citace: 1. květen 2011.] Dostupné z: <http://msdn.microsoft.com/en-us/library/ms243147.aspx>.
12. **Janošík, Dušan.** Vyvojar.cz - Něco málo k technologiím WPF a Silverlight. [Online] 13. květen 2007. [Citace: 1. květen 2011.] Dostupné z: <http://www.vyvojar.cz/Articles/473-neco-malo-k-technologiim-wpf-a-silverlight.aspx>.
13. MSDN Library - WPF Performance Suite. [Online] [Citace: 1. květen 2011.] Dostupné z: <http://msdn.microsoft.com/en-us/library/aa969767.aspx>.
14. MSDN Blogs - Performance characteristics of the Silverlight DataGrid. [Online] 13. duben 2011. [Citace: 1. květen 2011.] Dostupné z: <http://blogs.msdn.com/b/mcsuksoldev/archive/2010/04/13/performance-characteristics-of-the-silverlight-datagrid.aspx>.